ERROR RESILIENT SCALABLE AUDIO CODING (ERSAC) FOR MOBILE APPLICATIONS

Jianping Zhou[†], Qian Zhang[†], Zixiang Xiong[‡], and Wenwu Zhu[†]
[†]Microsoft Research China, 49 Zhichun Road, Beijing, 100080, P. R. China
[‡]Dept. of Electrical Engineering, Texas A&M University, College Station, TX 77843
{i-jpzhou,qianz,wwzhu}@microsoft.com; zx@lena.tamu.edu

ABSTRACT – Delivering high-fidelity audio over wireless channels is a challenging task because the wireless channel presents not only erasure errors, but also random bit errors. These errors have severe adverse effect on decompressing the received audio bitstream and may crash the decoder completely. To solve this problem, we propose error resilient scalable audio coding (ERSAC) for mobile applications by performing data partition and reversible variable length coding in the scalable audio bitstream. Simulation results show that ERSAC has very effective error-resilience in addition to bitstream scalability.

1. INTRODUCTION

With the advent of the Internet age, streaming high-fidelity audio has become a reality. It is thus natural to extend audio streaming to wireless communications so that mobile users can listen to music from handheld devices. However, delivering audio over wireless channels is a very challenging task because the wireless channel presents not only erasure errors caused by large-scale path loss and fading, but also random bit errors due to the wireless connection. These bit errors have severe adverse effect on decompressing the received bitstream. If not handled properly, they will crash the decoder. To combat these errors, forward error correction (FEC) has been used for protecting the compressed data [1] [2] [3] [4]. However, no matter how carefully the data are protected before transmission, the received data may still have bit errors. Therefore, studying error resilience in audio coding is necessary for audio over wireless channels to overcome the random bit errors. While there has been work done on error resilient video coding [5] [6], to the best of our knowledge, there is no report on error resilient audio coding in the literature. Error resilient schemes for video coding cannot be directly ported to audio coding because the characteristics of audio and video are different. There exists strong correlation between adjacent video frames that can be exploited to recover the data corrupted in the transmission [5] [6]. In contrast, there is almost no correlation between adjacent audio frames in the time domain. Moreover, audio coding artifacts caused by corrupted frames are very annoying to human ears. In this work, we propose error resilient scalable audio coding (ERSAC), in which we employ data partition and reversible variable length codes (RVLC) for scalable audio coding. Data partition is applied to limit the error propagation between different data segments; while RVLC are used to locate errors and minimize error propagation.

2. SCALABLE AUDIO CODING

Scalable audio coding has become increasingly popular recently [4] [7] [8] because it can efficiently accommodate the bandwidth fluctuation. A scalable audio bitstream typically consists of a base layer plus a number of enhancement layers. It is possible to use only a subset of the layers to decode the audio with lower sampling resolutions and/or quality. In streaming applications, the layers in a scalable bitstream are selectively delivered to adapt to network bandwidth fluctuation and packet loss level. For example, when the available bandwidth is low or packet loss ratio is high, only the base layer is transmitted.

In our original scalable audio codec, the audio signal is first split into individual time segments, which are filtered by a polyphase quadrature filter and grouped into four subbands to facilitate scalability in sampling resolution. The modified DCT (MDCT) is then performed on each subband and the resulting MDCT coefficients weighted by a psychoacoustic mask function. Finally, each weighted subband is encoded into an embedded bitstream using bit-plane coding, where each bit plane is coded into one layer or data unit (DU).

3. ERROR RESILIENT SCALABLE AUDIO CODING

3.1 Data Partition

Each DU in the original audio bitstream consists of significance/sign bits, followed by refinement bits. See Fig. 1, where necessary dummy zeros are added for the byte-alignment. The sign bits and refinement bits are not entropy coded, hence bit errors among them will not propagate. In contrast, the significance bits are compressed with variable length codes (VLC). When an error occurs in this portion of the bitstream, it will propagate and the whole DU will be damaged, including the sign data as well as the refinement data. DU multiplexing makes this situation more complex because when the decoder detects an error, it does not know the exact error location. As a result, the whole DU has to be discarded, no matter where the error occurs.

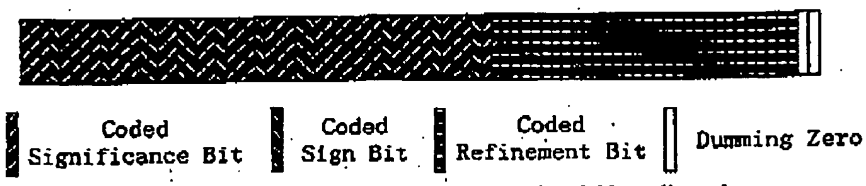


Figure 1. Bitstream syntax in the original scalable audio coder.

In our proposed ERSAC scheme, we first deinterleave the significance data, the sign data, and the refinement data and put them in three independent partitions. Thus, any error in the DU can be isolated and restricted to a particular partition. To locate errors among different partitions, the decoder must know the partition boundaries. However, this is impossible without *a priori* side information. So we put the refinement bits before other bits. This way the decoder can deduce the size of the refinement data from the DUs in previous layers. This resolves the ambiguity about the refinement partition. To finish the job, we use significance/sign boundary mark (SBM) to distinguish the significance partition from the sign partition. Because the VLC used in our scheme have a finite code tree, we can pick the SBM as an invalid codeword. In addition, for error robustness reasons, we choose the SBM to be sufficiently far in Hamming distance from other codewords so that it can be detected even if it is corrupted. The new data structure in ERSAC is depicted in Fig. 2. The length of the SBM is two or three bytes, which is the only overhead caused by ERSAC. It is very small and can be ignored, compared to the length of the DUs.

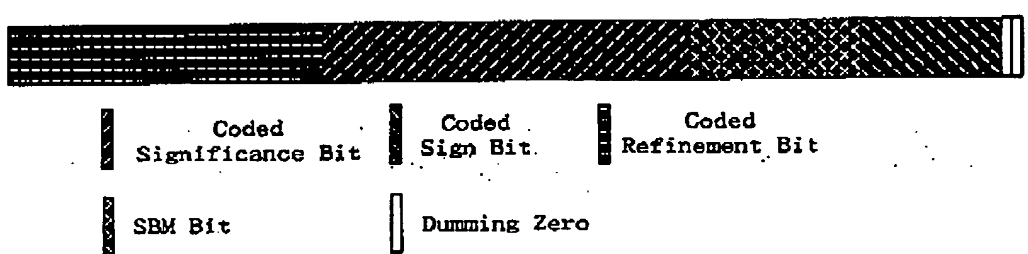


Figure 2. Data structure in our proposed ERSAC scheme.

3.2 Reversible Variable Length Codes

Reversible variable length codes (RVLC) are special VLC that can be decoded instantaneously both in the forward and backward directions [11]. When bit errors occur, the decoder can locate them by comparing the decoded results in the two different directions. Owing to this, RVLC have received significant attention recently [12] and been applied widely to error resilient video coding [13]. Generally, the two-way decoding property of RVLC will reduce the coding efficiency. However, there are some special types of RVLC that allow two-way decoding while retaining the efficiency of traditional (non-reversible) VLC. Reversible exponential Golomb (Exp-Golomb) codes [10] belong to this category. They were originally proposed in [12] as an extension of the Exp-Golomb codes [9]. Their length distribution is identical to the Exp-Golomb codes. Therefore, they can increase the robustness to channel errors while suffering no loss in coding efficiency.

Like Golomb codes, Exp-Golomb codes are associated with an order that is small for coding low entropy sources and large for coding high entropy sources. For binary sources, the optimal order can be calculated by the probability of zero. Depending on the order, each codeword consists of a variable-length prefix and a fix-length suffix. Exp-Golomb Codes is not sensitive to the value of the order and the range of the order is very limited. Hence, it is easy to choose a suitable order. In our experiments, the order is determined by the property of the significance bits in bit-plane coding. We set it to one in the first two bit-planes and two in other bit-planes.

Reversible Exp-Golomb codes are applied to the significance bits in our ERSAC scheme. Note that the codewords have a finite code tree. Some nodes on the code tree are thus invalid and can be serves as "traps" to detect errors. Once the decoder encounters an invalid codeword, it knows that errors exist in the bitstream, although it does not know the exact locations. Normally the received significance data are decoded both in the forward and backward directions. In case of an error, the decoder will locate it from either the forward or the backward decoding pass.

3.3 Error Handling

Because errors in the sign and refinement bits are non-propagating, our main task is to detect propagating errors in the significance bits coded by reversible Exp-Golomb codes. Due to data partition and the SBM, the boundary of the significance data can be known in advance. RVLC can then be used to track and locate the errors. Normally the significance data is decoded both in the forward and backward directions. When an error (e.g., an invalid codeword) is detected, the reversible Exp-Golomb decoder will stop and locate it in either decoding direction. Furthermore, one can apply sanity checks on the decoded significance bits because the number of the significance bits is known before decoding and the number of binary ones in them must be identical to the number of sign bits. If no errors are detected in both the forward and backward decoding directions and the decoded data pass the sanity check, the decoding result is declared to be correct. If error happens in decoding, the results of two decoders will be compared and identical portions in the two decoded versions are considered to be correct. By this means, most potentially correct bits can be utilized in the subsequent source decoding stage.

4. SIMULATION RESULTS

Extensive simulations are carried out to test the performance of our proposed ERSAC scheme. The MPEG-4 standard audio clips horn23_2, trpt21_2 and vioo10_2 are used. The scalable audio coder encodes each audio clip at a fixed rate of 64 kbps. Two simulated wireless network conditions are picked: one using the Gilbert model with different BERs and the other one having different BERs with Rayleigh fading. Note that such network conditions are obtained after L1, L2 and L3 of wireless network and are very typical in mobile applications. We assume that no channel coding is applied to the scalable bitstream and that the header information is well protected. We illustrate the quality of the decoded audio through the noise-mask-ratio (NMR), where a lower value of the NMR shows a better quality decoded audio. In the simulations, our ERSAC scheme is compared to the original scheme -- referred to as SAC for simplicity.

In our first set of experiments, we simulate a wireless network environment using the Gilbert model with channel fading length of 4 bits at the link layer. The BERs at the application layer are 10e-3 and 10e-4, respectively. The NMRs of SAC and ERSAC under these BERs are shown in Fig. 3. There are some bursty peaks on the curves for SAC, showing the occurrence of bit errors, but none on the curves for ERSAC. Numerical NMR results from streaming three audio clips (homn23_2, trpt21and vioo10_2) are shown in Table 1. These results are the average values over 10 runs because of the randomness of the simulated channel.

Table 1. Average NMR results for multiple audio clips with different BERs

Audio Clips	Horn23_2		Trpt21_2		Vioo10_2	
	BER = 10e-3	BER = 10e-4	BER = 10e-3	BER = 10e-4	BER = 10e-3	BER = 10e-4
SAC	4.7804	2.8476	4.6082	2.9369	4.7708	2.9483
ERSAC	4.0961	2.5872	4.3261	2.6058	4.2713	2.1820

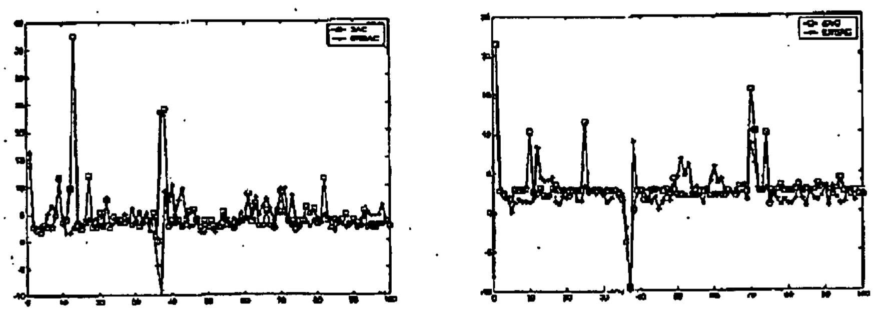
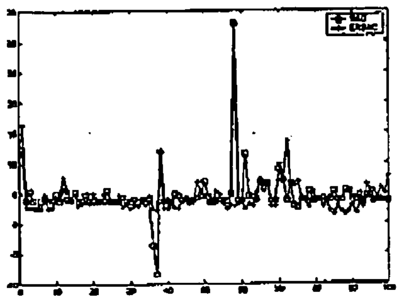


Figure 3. NMRs of SAC and ERSAC with BER being 10e-3 (left) and 10e-4 (right). The audio clip is horn23_2 and the bandwidth is 64 kbps.

In our second set of experiments, we simulate a wireless network environment with Rayleigh fading. The parameters are: Eb/No=22dB for BER=1e-3 and Eb/No=32dB for BER=1e-4. The Doppler spread is 5.27Hz, which is computed from $f_d = vf_c / c$, with the walking velocity v=3 km/h (or 0.83m/s), the carrier frequency f_c =1900MHZ, and light velocity $c=3\times 10^8$ m/s. The NMRs of SAC and ERSAC under BERs of 10e-3 and 10e-4 are shown in Fig. 4. Again, there are some

burst peaks on the SAC curves, but almost none no the ERSAC curves. Average NMR results from streaming honrn23_2, trpt21and vioo10_2 under constant network bandwidth (64kbps) and different BERs are shown in Table 2.



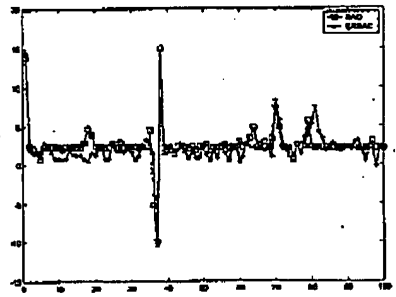


Figure 4. NMRs of SAC and ERSAC with BER being 10e-3 (left) and 10e-4 (right). The audio clip is horn23_2 and the bandwidth is 64 kbps.

Table 2. Average NMR results for multiple audio clips with different BERs

Audio Clips	Horn23_2		Trpt21_2		Vioo10_2	
	BER = 10e-3	BER =10e-4	BER = 10e-3	BER = 10e-4	BER = 10e-3	BER = 10e-4
SAC	4.5713	2.6185	4.4139	2.7738	4.4192	3.0124
ERSAC	4.2045	2.1698	4.1738	2.1050	4.0264	2.1757

These results show that ERSAC indeed improves the error resilience of scalable audio coding and is more immune to bit errors than SAC. Subjective tests are also conducted. Listeners perceive a better quality of the delivered audio given by ERSAC, while annoying artifacts are audible from SAC.

5. REFERENCE

[1] C. Perkins, O. Hodson, and V. Hardman, "A survey of Packet Loss Recovery Techniques for Streaming Audio", IEEE Trans. Network, vol. 12, pp. 40-48, Sept. 1998.

[2] M. Podolsky, C. Romer, and S. McCanne "Simulation of FEC-based error control for packet audio on the Internet," Proc. INFOCOM '98, vol. 2, pp. 505-515, 1998.

[3] N. Shacham and P. McKenney, "Packet recovery in high-speed networks using coding and buffer management", Proc. INFOCOM '90, vol. 1, pp. 124-131, 1990.

[4] T. Moriya, T. Mori, N. Iwakami, and J. Akio, "A design of error robust scalable coder based on MPEG-4/Audio," Proc. ISCAS '2000, vol. 3, pp. 213-216, 2000.

[5] J. Lu, K.Letainef and M. Liou, "Robust Video Transmission over Correlated Mobile Fading Channels," IEEE Trans. On Circuits and Systems for Video Technology, vol. 9, pp. 737-750, Aug. 1999.

[6] N. Farber, B. Girod and J. Villasenor., "Extensions of ITU-T Recommendation H.324 for Error-Resilient Video Transmission," IEEE Communications Magazine, p. 120-128, June 1998.

[7] T. Moriya, N. Iwakami, J. Akio, and T. Mori, "A design of lossy and lossless scalable audio coding," Proc. ICASSP'00, vol. 2, pp. 889-892, 2000.

[8] P. Kudumakis and M. Sandler, "Wavelet packet based scalable audio coding", Proc. ISCAS'96, vol. 2, pp 41-44, 1996.

[9] S. W. Golomb, "Run-length encodings," IEEE Trans. Inform. Theory, vol. IT-12, pp. 399-401, July 1966.

[10] J. Teuhola, "A Compression Method for Clustered Bit-vectors," Information Processing Letters, vol. 7, pp. 308-311, October 1978.

[11] T. Takishima, M. Wada, and H. Murakami, "Reversible Variable Length Codes," IEEE Trans. Comm., vol. 43, No.2/3/4, pp158-162, 1995.

[12] G. Wen and J. Villasenor, "A class of Reversible Variable Length Codes for Robust Image and Video Coding," Proc. ICIP '97, vol. 2, pp. 65-68, Santa Barbara, CA, Oct. 1997.

[13] R. Talluri, "Error Resiliency in ISO's MPEG-4 Video Coding Standard," IEEE Commu. Mag, vol. 36, pp. 112-119, June 1998.